

1. ¿En que crees que se diferencian un programa y un proceso?

Programa: Es una serie de comandos ejecutados por el equipo.

Proceso: Un programa en ejecución es un proceso.

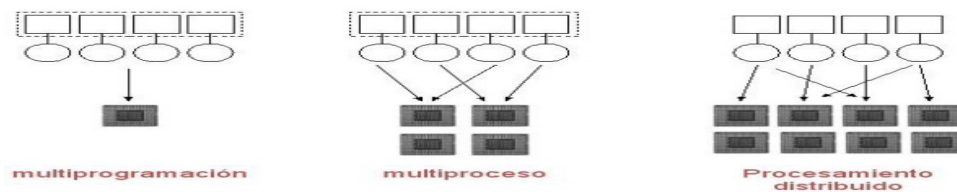
2. ¿Crees que la concurrencia software implica paralelismo hardware?

La concurrencia es la capacidad del CPU para procesar más de un proceso al mismo tiempo, Pero que no implica esto realmente al paralelismo hardware. Para comprender esto es necesario entender cómo funciona un procesador. Un procesador puede procesar al mismo tiempo el mismo número de procesos que el número de CORES que tiene, de esta forma, si un procesador tiene un CORE, entonces solo podrá ejecutar un proceso a la vez, por otro lugar, si tenemos 8 CORES, entonces podremos ejecutar hasta 8 procesos al mismo tiempo.

3. ¿Dónde surge la concurrencia?

Concurrencia es la tendencia de las cosas a producirse al mismo tiempo en un sistema. La concurrencia es un fenómeno natural, por supuesto. En el mundo real, en un momento dado, suceden muchas cosas de forma simultánea. Cuando se diseña software para supervisar y controlar los sistemas del mundo real, hay que enfrentarse a esta concurrencia natural.

4. Observando las siguientes imágenes, ¿podrías deducir las características más importantes de cada uno de los diferentes entornos hardware donde surge la concurrencia?



Un procesador puede procesar al mismo tiempo el mismo número de procesos que el número de CORES que tiene, de esta forma, si un procesador tiene un CORE, entonces solo podrá ejecutar un proceso a la vez, por otro lugar, si tenemos 8 CORES, entonces podremos ejecutar hasta 8 procesos al mismo tiempo. Y si podemos ejecutar 2 o más procesos paralelo entonces se surge la concurrencia.

5. ¿Qué entiendes por sistema inherentemente concurrente y por sistema potencialmente concurrente?

Sistemas inherentemente Concurrente: El entorno que tiene actividades simultanea.
p.eg: un Horario

Sistemas Potencialmente Concurrente: No es necesario la concurrencia, pero para aumentar la velocidad se puede sacar partido de ella.
p.eg: Para aumentar la velocidad de ejecución.

6. ¿Qué diferencias y relación crees que existen entre ellos?

Los hilos se distinguen los procesos en que los procesos son independientes, los procesos llevan bastante información de estados y interactúan con el mecanismo de comunicación por el sistema.

Por otro lugar muchos hilos comparten recursos directamente. En muchos de los sistemas operativos que proveen facilidades para los hilos que cambian de un proceso a otro. Y los hilos comparten datos y espacios de direcciones mientras que los procesos no lo hacen.

7. ¿Qué entiendes por programación concurrente?

Un único programa ejecutado simultáneamente de múltiples tareas, que pueden ser de un conjunto de procesos o hilos.

8. ¿Se te ocurre algún ejemplo de programa multihilo?

Los editores de video y audio.

9. ¿Crees que la programación concurrente tiende a desaparecer o que, por el contrario, cada vez adquiere un papel más relevante?

Programación concurrente no se tiende a desaparecer porque Mejora el aprovechamiento, La velocidad de ejecución, Soluciona de problemas de naturaleza concurrente (sistemas de control, tecnologías web, Aplicaciones basadas en interfaces de usuarios.), Facilita la programación (Tareas se pueden estructurar en procesos distintos) y Ventajas derivadas del procesamiento asíncrona (p.eg Writer de libre office guarda un fichero mientras se continua con la escritura por el usuario sin interferir en la misma)

10. ¿Cómo crees que se expresa la concurrencia en un programa?

Existen dos técnicas para expresar la concurrencia:

De forma Manual: Mediante un lenguaje concurrente que puede ser por ejemplo: java

De forma Automática: El S.O se encarga automáticamente por ejemplo : multiprogramación.

11. ¿Conoces algún lenguaje de programación concurrente?

Java, python, Ada etc.

En un programa multihilo,

12. ¿el flujo de ejecución es siempre el mismo?

En un programa Secuencial: En un orden total de las instrucciones no es lo mismo.

Y en un programa Concurrente: Ante un orden parcial de ejecución no se puede saber cuál va a ser el flujo de ejecución.

13. ¿se pueden obtener resultados diferentes ante un mismo conjunto de datos de entrada?

No se puede obtener resultados diferentes ante un mismo conjunto de datos de entrada.

14. Si es así, ¿significaría esto que hay algún tipo de error en el programa? Pon un ejemplo

Sí, si en el programa esta puesto aleatorio o que cambie el valor dentro del programa cada vez que ejecute, pues claro que va a dar diferente resultado. (Depende del programa)

15. ¿Puedes enumerar beneficios de la programación concurrente?

Mejora el aprovechamiento, La velocidad de ejecución, Soluciona de problemas de naturaleza concurrente (sistemas de control, tecnologías web, Aplicaciones basadas en interfaces de usuarios.), Facilita la programación(Tareas se pueden estructurar en procesos distintos) y Ventajas derivadas del procesamiento asíncrona (p.eg Writer de libre office guarda un fichero mientras se continua con la escritura por el usuario sin interferir en la misma)

16. ¿Crees que la programación concurrente añade más dificultad o complejidad a la tarea de programar? Si es así, comenta ejemplos en los que esta complejidad se ponga de manifiesto.

La programación concurrente no se añade más dificultad o complejidad a la tarea sino que se hace más fácil.

17. Enumera dos casos de sistemas potencialmente concurrentes en los que nos beneficiaríamos de la utilización de la concurrencia.

El corrector del documento mientras escribes.